A Fully Distributed Spatial Index for Wireless Data Broadcast *

Wang-Chien Lee
Pennsylvania State University
wlee@cse.psu.edu

Baihua Zheng
Singapore Management University
bhzheng@smu.edu.sg

Abstract

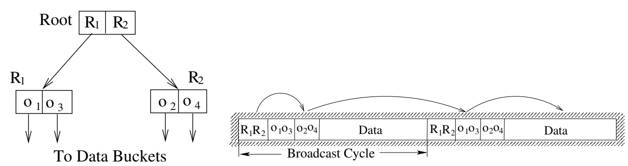
To support location-based services in wireless data broadcast systems, a distributed spatial index (called DSI*) is proposed in this paper. DSI is highly efficient because it has a linear yet fully distributed structure that naturally facilitates multiple replications of the index by sharing links in different search trees. Search algorithms for point queries, window queries, and kNN queries, based on DSI are presented. Empirical evaluation of DSI are conducted. Result shows that DSI significantly out-performs R-tree and Hilbert Curve Index, two state-of-the-art spatial indexing techniques for wireless data broadcast.*

1. Introduction

Information is important to users, yet it is only valuable when available at the right time, *right place*. Compared with the conventional point-to-point connection, wireless broadcast due to its scalability is a very attractive approach to disseminate information to mobile users in the upcoming pervasive computing era. In this paper, we propose a novel spatial index, called *Distributed Spatial Index (DSI)*, in support of location-based spatial queries issued from mobile users in wireless data broadcast systems.

Many spatial index structures have been proposed for accessing spatial data, including R-tree, KD-tree, Quad-tree, etc. Among those, R-tree is the most well received for its simplicity and ability to handle a variety of spatial data and queries [1]. A search algorithm based on R-tree typically expands the search space around the query point using a branch-and-bound approach. Consequently, the navigation order of R-tree is dynamically determined based on the position of the query point, which results in backtracking. Thus, R-tree is better supported by random access storages, such as memory and disk.

In a wireless broadcast channel, however, data objects are broadcast based on a pre-defined sequence (called a *broadcast program*) and thus an object is only available when it is on the air. Consequently, search algorithms designed based on random access may incur a significant access latency. Figure 1 depicts an example. Assuming that a search algorithm first visits the node R_2 and then R_1, after visiting the root node, while the server broadcasts nodes in the order of root, R_1, and R_2. Consequently, if a client wants to visit node R_1 after it retrieves R_2, it will have to wait until the next cycle because R_1 has already been broadcast. This significantly extends the access latency and it occurs every time a search order is different from the broadcast order. Thus, new index structures and search algorithms need to be developed to fit the sequential access property of wireless data broadcast.



(a) R-tree Index (b) Branch-and-Bound Search

Figure 1. Linear Access on Broadcast Channel

2. Distributed Spatial Index (DSI)

Distributed spatial index (DSI) is designed to meet the constraints of wireless data broadcast systems. The main idea is to allow a client to start query processing as soon as possible in order to minimize the access latency while still conserving tuning time. DSI distributes the index information over the whole broadcast cycle and equips a client, no matter when it switches on to the channel, with sufficient information to conduct the spatial search.

Taking into account the sequential access property of wireless broadcast, Hilbert Curve (HC) is adopted in DSI to determine broadcast order of data objects. HC is a space-filling curve, which crosses every point in a grid exactly once without crossing itself. Figure 2 shows an HC of order 3. The numeric labels represent the positions of the objects in terms of *HC values*. For instance, point $(1, 1)$ has the HC value of 2. Therefore, HC provides a linear order of the objects within a non-linear space. Since each object in the original space has a deterministic position along the HC, i.e., the

* This work was supported in part by the National Science Foundation under Grant IIS-0328881

HC value of an object is unique, the corresponding HC values to the objects can serve as the index key.

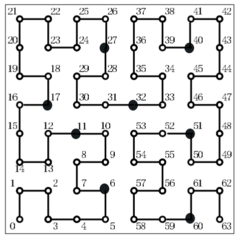


Figure 2. Hilbert Curve of Order 3

By default, data objects are broadcast in the ascending order of their HC values. The basic idea of DSI is to divide the whole set of data objects into n_F *frames* and associate with each frame an *index table*. The index table maintains information regarding to the HC values of data objects to be broadcast with specific waiting interval from the current packet.

A DSI table consists of a number of table entries, τ_i, in the form $\langle HC'_i, P_i \rangle$, where $0 \leq i \leq (log_r(n_F) - 1)$, r is a selected exponential base, and n_F is the number of the frames within one broadcast cycle. Note that, logically, the set of frames starting from any arbitrary frame F until the frame before reappearance of F forms a broadcast cycle. Therefore, the index table associated with a frame F is designed to cover the next $(n_F - 1)$ frames following F, i.e., providing index information on HC values of all the frames within a broadcast cycle. Grouping those n_F frames as a set, P_i points to the next r^ith frame. HC'_i is the smallest HC value of the objects within the frame pointed by P_i. Thus, the ith entry of the index table provides range information of HC values amongst data objects in the r^ith to $(r^{i+1} - 1)$th frames. In other words, the number of frames covered is exponentially increased with the order of index table entries.

If we assume that each frame contains one object, and the index base is two, Figure 3 shows the broadcast of data objects based on the example in Figure 2 and the index tables. In this example, $n_F = 8$ and thus each index table has 3 entries. The DSI tables corresponding to frames of data objects O_6 and O_{32} are also shown in the figure. Take the index table for frame O_6 as an example: τ_0 contains a pointer to the next upcoming frame whose HC value is 11, τ_1 contains a pointer to the second frame with HC value 17, and the last entry τ_2 points to the fourth frame.

Efficient search algorithms are proposed to answer point queries, window queries, and KNN search based on DSI. For point queries, the HC value of the query point is first obtained. Thereafter, the client compares this HC value to the values shown in the index table and jumps towards the target. For window queries, the entire HC is partitioned into several segments based on the intersections of query window and HC. Each segment is either fully inside or outside the window. Guided by all the inside segments, clients can check DSI and follow the pointers that direct the jump to the frames containing answers. KNN search is relatively complicated. The basic idea behind our kNN algorithms is to determine a search space based on the partial knowledge of object distribution obtained from index table. The search space will continuously shrink as more knowledge of the data distribution is obtained. The search can be terminated when the search range could not be shrink and all the objects inside the current search ranges are received.

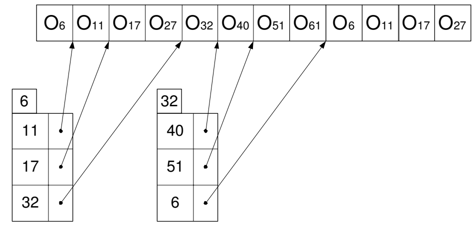


Figure 3. DSI for the Running Example

We evaluate the performance of DSI by comparing it with two state-of-the-art spatial indexing techniques for wireless data broadcast, i.e., R-tree and Hilbert-Curve index [1, 3]. In this study, R-tree and Hilbert-Curve index are distributed among the wireless broadcast channel based on the well-known distributed indexing scheme [2]. Both analytical data and real data are employed in the simulation. The results show that DSI outperforms both R-tree and Hilbert-Curve index significantly.

3. Conclusion

This paper exploits issues for supporting spatial queries in wireless data broadcast systems. A fully distributed spatial index, called *DSI*, is proposed. DSI naturally replicates multiple search trees into a linear structure and fully distributes this index structure over the whole broadcast cycle. As a result, it allows a search to start immediately. Our next step is to look into prototype of a wireless data broadcast system that supports spatial queries. We are also interested in further exploiting the research issues in unreliable wireless data broadcast environments.

References

[1] A. Guttman. R-trees: A dynamic index structure for spatial searching. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pages 47–54, 1984.

[2] T. Imielinski, S. Viswanathan, and B. R. Badrinath. Data on air - organization and access. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 9(3), May-June 1997.

[3] B. Zheng, J. Xu, W. C. Lee, and D. L. Lee. Energy-conserving air indexes for nearest neighbor search. In *Proceedings of the 9th International Conference on Extending Database Technology (EDBT'04)*, Heraklion - Crete, Greece, March 2004.